

25X1A

REPORT NO.

CD NO.

DATE DISTR. 27 March 1950

NO. OF PAGES 1

NO. OF ENCLS.
(LISTED BELOW)

SUPPLEMENT TO
REPORT NO.

```

1  # Import the necessary libraries
2  import pandas as pd
3  import numpy as np
4  from sklearn.preprocessing import StandardScaler
5  from sklearn.model_selection import train_test_split
6  from sklearn.metrics import mean_squared_error, r2_score
7  from sklearn.linear_model import LinearRegression
8  from sklearn.ensemble import RandomForestRegressor
9  from sklearn.svm import SVR
10 from sklearn.neighbors import KNeighborsRegressor
11 from sklearn.tree import DecisionTreeRegressor
12 from sklearn.metrics import mean_absolute_error
13
14 # Load the dataset
15 data = pd.read_csv('data.csv')
16
17 # Split the data into features and target variable
18 X = data[['temp', 'casual', 'registered']]
19 y = data['rental_rate']
20
21 # Split the data into training and testing sets
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
23
24 # Standardize the features
25 scaler = StandardScaler()
26 X_train = scaler.fit_transform(X_train)
27 X_test = scaler.transform(X_test)
28
29 # Train the Linear Regression model
30 lr = LinearRegression()
31 lr.fit(X_train, y_train)
32
33 # Predict the rental rate using the Linear Regression model
34 y_lr_pred = lr.predict(X_test)
35
36 # Calculate the Mean Squared Error (MSE) and R-squared (R2) for the Linear Regression model
37 mse_lr = mean_squared_error(y_test, y_lr_pred)
38 r2_lr = r2_score(y_test, y_lr_pred)
39
40 # Train the Random Forest Regressor model
41 rf = RandomForestRegressor()
42 rf.fit(X_train, y_train)
43
44 # Predict the rental rate using the Random Forest Regressor model
45 y_rf_pred = rf.predict(X_test)
46
47 # Calculate the Mean Squared Error (MSE) and R-squared (R2) for the Random Forest Regressor model
48 mse_rf = mean_squared_error(y_test, y_rf_pred)
49 r2_rf = r2_score(y_test, y_rf_pred)
50
51 # Train the Support Vector Regression (SVR) model
52 svr = SVR()
53 svr.fit(X_train, y_train)
54
55 # Predict the rental rate using the SVR model
56 y_svr_pred = svr.predict(X_test)
57
58 # Calculate the Mean Squared Error (MSE) and R-squared (R2) for the SVR model
59 mse_svr = mean_squared_error(y_test, y_svr_pred)
60 r2_svr = r2_score(y_test, y_svr_pred)
61
62 # Train the K-Nearest Neighbors (KNN) model
63 knn = KNeighborsRegressor()
64 knn.fit(X_train, y_train)
65
66 # Predict the rental rate using the KNN model
67 y_knn_pred = knn.predict(X_test)
68
69 # Calculate the Mean Squared Error (MSE) and R-squared (R2) for the KNN model
70 mse_knn = mean_squared_error(y_test, y_knn_pred)
71 r2_knn = r2_score(y_test, y_knn_pred)
72
73 # Train the Decision Tree Regressor model
74 dt = DecisionTreeRegressor()
75 dt.fit(X_train, y_train)
76
77 # Predict the rental rate using the Decision Tree Regressor model
78 y_dt_pred = dt.predict(X_test)
79
80 # Calculate the Mean Squared Error (MSE) and R-squared (R2) for the Decision Tree Regressor model
81 mse_dt = mean_squared_error(y_test, y_dt_pred)
82 r2_dt = r2_score(y_test, y_dt_pred)
83
84 # Print the results
85 print('Linear Regression: MSE =', mse_lr, 'R2 =', r2_lr)
86 print('Random Forest Regressor: MSE =', mse_rf, 'R2 =', r2_rf)
87 print('Support Vector Regression: MSE =', mse_svr, 'R2 =', r2_svr)
88 print('K-Nearest Neighbors: MSE =', mse_knn, 'R2 =', r2_knn)
89 print('Decision Tree Regressor: MSE =', mse_dt, 'R2 =', r2_dt)

```

- This document is hereby regraded to
CONFIDENTIAL in accordance with the
letter of 18 October 1978 from the
Director of Central Intelligence to the
Archivist of the United States.
Next Review Date: 2038

Document No. 10
No Change in Class
☒ Declared
Class. Group 1: 15 5 0
Auth: HR 70-2
Date: 0-6 JUN 1978 By: 28

CONFIDENTIAL

~~SECRET/CONTROLLING~~ ~~WARNING NOTICE: THE DISTRIBUTION LISTING MUST BE~~

DISTRIBUTION

EXCISED BEFORE PUBLIC RELEASE OF THIS DOCUMENT

25X1A